Kindly amend this application as indicated herein.

IN THE SPECIFICATION:

Kindly amend the specification at the Title and at paragraphs [0032], [0064], and [0067] as follows:

Title:

Fixed Length Memory to Memory ~~Instruction Set~~ Arithmetic and Architecture for a Communications Embedded Processor System

[0032] Figure 1B is a flowchart illustrating the processes performed by processor 100. It includes the steps of loading 132 data to the register file 120, performing 134 computations on the data, and storing 136 data back to memory 110. As can be seen from Figure 1B, the RISC processor 100 has a "load and store" architecture. Data from memory 110 cannot be accessed directly. The data from memory 110 is first loaded 132 into the register file 120. The ALU 130 only obtains data from the register file 120. The width of the arithmetic in the ALU 130 is 32 bits. The width of data from the register file 120 is also 32 bits. The width of the data from the memory 110 may be variable. If the width of the data from the memory 110 is 8 or 16 bits, the sign extender 115 is used to extend the data to 32 bits before it is stored in the register file 120. The ALU 130 then performs 134 the requisite computations on the data obtained from the register file 120, and the result is stored in the register file. This result can then be stored 136 from the register file 120 back to memory 110. When the result is stored 136 in memory, a truncator 125 may be used to store the result as 8-bit or 16-bit data. [[Alternately]] Alternatively, the result may be stored as [[a]] 32-bit data in the memory 110.

[0064] It can be seen that in one embodiment, the present invention does not inherently have a direct memory addressing mode. (In such an address mode, the address of the memory in which the first operand is located is directly specified.) However, in one embodiment of the present invention, ~~however,~~ such a direct memory addressing mode can be used by inserting an immediate into an address register. This is done using the Move Address register Immediate (MoveAI) instruction. In particular, by using the MoveAI instruction in conjunction with the Address Register + 7-bit Immediate instruction, a large number of bits can be used to specify the memory address. In one embodiment, a 24-bit Immediate can be moved into the Address Register. This address register can then be used in conjunction with the Address Register + 7-bit Immediate instruction, to obtain a 31-bit memory address.

[0067] Row 2 in Figure 6 corresponds to the Register + Register addressing mode discussed above. One of these registers is selected from the 8 address registers in the address register file 512, while the other is selected from the 32 data registers in the general ~~data~~ register file 514.